REMARKS

This is in response to the Final Office Action mailed 2/7/2007, and further in view of the Request for Continued Examination (RCE) submitted herewith.

Applicants have cancelled previously pending claims 1 through 22. Claims 23-44 have been newly added with respect to the RCE concurrently filed herewith. Applicants wish to emphasize that they are not conceding in this response that those claims are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are only for facilitating expeditious prosecution. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications.

This response should obviate outstanding issues and make the pending claims allowable. Reconsideration of this application is respectfully requested in view of this response.

STATUS OF CLAIMS

Previously pending claims 1-22 are cancelled.

Claims 23-44 have been newly added with respect to the RCE concurrently filed herewith.

OVERVIEW OF CLAIMED INVENTION

The present invention provides for an extensible identification system for nodes in a hierarchy, wherein each node is assigned a concatenation of decimal based values. The identification value uniquely identifies the node, provides an order for the node, and identifies its

parent, child, and sibling relationships with other nodes. Also, the IDs assigned can be encoded to be byte comparable. Furthermore, the IDs assigned to nodes need not be modified when changes (adding/deleting a child node or a subtree of nodes) are made in the hierarchy. Additionally, in the event of such a change, the order and relationships between the parent, child, and sibling nodes are retained.

The present invention provides for a robust method for updating a computer-stored hierarchical structure of nodes via a node identification technique, wherein the method comprises the steps of: (a) receiving an instruction to insert a new node at an insertion point in a computer-stored hierarchical structure; (b) identifying one of, or a combination of the following: a left node ID value closest to the left of the insertion point or a closest right node ID value closest to the right of the insertion point; (c) calculating a new ID value based upon node ID value(s) identified in (b), said calculated value greater than ID values of nodes to the left of said insertion point and less than ID values of nodes to the right of said insertion point, said new ID value based upon a low/high key value, said high key value representing positive infinity and said low key value representing negative infinity; and (d) updating the computer-stored hierarchical structure by inserting the new node in the hierarchy and associating the new node with the calculated ID value. As a result of such an implementation, the order, node ID values, and relationships between parent, child, and siblings in the hierarchical structure of nodes remain unchanged with the insertion of new nodes.

The present invention provides a way for assigning IDs to nodes in a hierarchy and provides many advantages, some of which include: (a) the IDs provide a way of ordering nodes

in a hierarchy; (b) the IDs describe a node's parent, child, and sibling relationships; (c) the IDs can be encoded such that they are byte comparable; (d) the IDs can be assigned to newly inserted nodes, anywhere in the hierarchy, and still maintain these properties; and (e) the IDs, once assigned, do not have to change even with changes to the hierarchy.

COMMENTS REGARDING O'NEIL ET AL. (USP 6,889,226)

O'Neil et al.'s Figures 5 and 6 show how data can be inserted (or "careted") into a hierarchical data structure. However, it should be noted that O'Neil's structure is restrictive in the fact that only odd numbers are used as position numbers for nodes.

The presently claimed invention can be distinguished from O'Neil because our system uses the notion of a positive infinity number 'x' and negative infinity number '0' to define the boundary of subtrees (see for example, Table 3 of the application-as-filed). For example, to insert between 1.1 and 1.2, we use the number 1.1.x.1. The x (positive infinity) is higher than any value that can represent any node within the subtree under 1.1. So under 1.1, the first child could be 1.1.1, second child is 1.1.2, third 1.1.3 and so on. But a child of 1.1 can never be equal to or greater than 1.1.x because x is higher than any value. Because of this, I can use the range $1.1 < a < 1.1.x$ to define the nodes within the subtree of 1.1.

The same argument applies to '0', where 0 represents negative infinity (which is used to go in the opposite direction).

O'Neil fails to teach or suggest such a robust scheme that uses the notion of high/low key values representing positive and negative infinity. Also absent in O'Neil's teachings are steps for concatenating a left node ID value with one or more high key values representing positive infinity or concatenating a left node ID value with one or more low key values representing negative infinity.

Absent such teachings, O'Neil can neither teach nor suggest many of the claimed features of the pending claims 23-44.

SUMMARY

As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of Applicants' presently claimed invention, nor renders them obvious. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

This response is being filed with a request for extension of time. The Commissioner is hereby authorized to charge the extension fee, as well as any deficiencies in the fees provided to Deposit Account No. 09-0460.

If it is felt that an interview would expedite prosecution of this application, please do not hesitate to contact Applicants' representative at the below number.

Respectfully submitted,

/ramrajsoundararajan/

Ramraj Soundararajan
Registration No. 53,832

IP Authority, LLC.
9435 Lorton Market Street #801
Lorton, VA 22039
571-642-0033

June 7, 2007